**ERROR’S IN PYTHON**

**1. WHAT IS SYNTAX ERROR?**

* The Python **SYNTAX ERROR** Occurs When the Interpreter Encounters Invalid Syntax in Code.
* When Python Code is Executed, The Interpreter Sends it to convert it into Bytecode.
* If the Interpreter finds any Invalid Syntax during the Compiling Stage, a **SYNTAX ERROR** is Thrown.

**Ex:**

a=22

print(a):

print(a):

        ^

SyntaxError: invalid syntax

**2. WHAT IS INDENTATION ERROR?**

* The **INDENTATION ERROR** can Occur When the Spaces or Tabs are not Placed Properly.
* There will not be an issue if the Interpreter does not find any issues with the Spaces or Tabs.
* If There is an Error due to Indentation, it will come in between the Execution and can be a Show Stopper.

**Ex:**

 a=22

print(a)

 a=22

^

IndentationError: unexpected indent

**3. What is TYPE ERROR?**

* If the both the Variables are different Data types then, it returns **TYPE ERROR.**
* The Python **TYPE ERROR** is an Exception that Occurs When the Data Type of an object is

is Inappropriate or Not Same.

* This can happen When an operation is Performed on an Object of an Incorrect Type, or

It is not supported for the Object.

**Ex:**

a=22

b='KIRAN'

c=a+b

print(c)

c=a+b

TypeError: unsupported operand type(s) for +: 'int' and 'str'

**4.What is VALUE ERROR?**

* If the Given Value is not present in a list then, it returns **VALUE ERROR.**
* The Python **VALUE ERROR** is an Exception that occurs when a Function receives an Argument of the Correct Data Type but an Inappropriate value.
* This Error usually occurs in Mathematical Operations that require a Certain Kind of Value.

**Ex:**

a=(11,22,33,44,55)

print(a.index(99))

print(a.index(99))

^

ValueError: tuple.index(x): x not in tuple

**5. What is INDEX ERROR?**

* If the given number is not present in a list then it returns **INDEX ERROR.**
* **INDEX ERROR** means list index Out of Range Error occurs in Python when an item from a list is Attempted to be Accessed that is outside the Index Range of the list.

**Ex:**

k=[11,22,33,44,55]

print(k[22])

print(k[22])

^

IndexError: list index out of range

**6. What is NAME ERROR?**

* If the Given Variable is not Declared then, it returns **NAME ERROR.**
* In Python, the **NAME ERROR** occurs when you try to use a Variable, Function, or Module that Doesn't Exist or wasn't used in a Valid Way.
* Some of the common mistakes that cause This Error are: Using a Variable or Function Name that is yet to be Defined.

**Ex:**

while i<=10:

    print(i)

    i=i+1

    while i<=10:

          ^

NameError: name 'i' is not defined. Did you mean: 'id'?

**7. What is KEY ERROR?**

* If the Given value is not present in a set, then it returns **KEY ERROR.**
* A Python **KEY ERROR** Exception is what is raised when you try to Access a Key that isn't in a Dictionary (dict()).
* Python's Official Documentation says that the **KEY ERROR** is raised when a Mapping key is Accessed and isn't found in the Mapping.
* A mapping is a Data Structure that Maps One Set of Values to Another.

**Ex:**

k={1:'SIVA',2:'KIRAN'}

print(k[7])

print(k[7])

        ^

KeyError: 7

**8. What is ATTRIBUTE ERROR?**

* If the Given Method (Function) is not present in the program then, it returns

**ATTRIBUTE ERROR**.

* The Python **ATTRIBUTE ERROR** is an Exception that occurs when an Attribute Reference or Assignment Fails.
* This can Occur when an Attempt is made to Reference an Attribute on a value that does not support the attribute.

**Ex:**

a=(11,22,33,44,55)

print(a)

a.remove(222)

print(a)

AttributeError: 'tuple' object has no attribute 'remove'

**9. What is a ZERO DIVISION ERROR?**

* A **ZERO DIVISION ERROR** is raised when you try to divide by 0. This is part of the **ARITHMETIC ERROR** Exception Class.

**Ex:**

a=22

b=0

c=a/b

print(a)

ZeroDivisionError: division by zero

**10. What is RECURSION ERROR?**

* A Python **RECURSION ERROR** Exception is raised when the Execution of your program Exceeds the Recursion limit of the Python Interpreter.
* Python: Maximum Recursion Depth Exceeded [How to Fix It] The maximum recursion depth in Python is 1000.

**Ex:**